Amendments to the Abstract of the Disclosure

Amend the Abstract of the Disclosure as follows:

~~The~~ An emulation system includes a clock generation logic for generating multiple asynchronous clocks, where each generated clock's relative phase relationship with respect to all other generated clocks is strictly controlled to speed up the emulation logic evaluation. Unlike statically designed emulator systems known in the prior art, the speed of the logic evaluation in the emulator need not be slowed down to the worst possible evaluation time since the clocking is generated internally in the emulator and carefully controlled. The emulation system does not concern itself with the absolute time duration of each clock, because only the phase relationship among the multiple asynchronous clocks is important. By retaining the phase relationship (and the initial values) among the multiple asynchronous clocks, the speed of the logic evaluation in the emulator can be increased. The RCC clock generation logic comprises a clock generation scheduler and a set of clock generation slices, where each clock generation slice generates a clock. The clock generation scheduler compares each clock's next toggle point from the current time, toggles the clock associated with the winning next toggle point, determines the new current time, updates the next toggle point information for all of the clock generation slices, and performs the comparison again in the next evaluation cycle. In the update phase, the winning slice updates its register with a new next toggle point, while the losing slices merely updates their respective registers by adjusting for the new current time. ~~The clock generation scheduler performs the following algorithm for each evaluation cycle:~~

~~(1) set initial values for all registers;~~

~~(2) from the current time, find the next toggle point for all the clocks;~~

~~(3) toggle the clock associated with this next toggle point;~~

~~(4) adjust the current time to be the time associated with this toggle point;~~

~~(5) adjust the next toggle point for the winning clock slice, while keeping all other clock slices' respective next toggle points (the toggle points will be the same for the losing slices but the time durations will be adjusted based on the new current time)~~.